Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1 – 47 (Cancelled).

48. (currently amended) A method for on demand functional verification of a software model of an application specific integrated circuit (ASIC), wherein said software model is written in a low-level programming language and wherein the method separately establishes ~~a~~ said software model ~~of the application specific integrated circuit to be generated~~ and functional verification tests to be applied to the software model ~~of the circuit~~ for constituting a verification platform comprising ~~a transmission~~an environment emulator debug mode and a verification mode, the method comprising:

creating, in the environment emulator debug~~transmission~~ mode, an autonomous circuit emulator ~~constituted with~~built around a data processing system, receiving in memory functional specification data obtained by replacing, at least by a language adaptation device, the software model ~~which is in a low level programming language and~~ that physically describes ~~an~~ the ASIC comprising a circuit under design to be validated, with a high level language abstract description, the autonomous circuit emulator generating response data structures in accordance with ~~a~~ said functional specification data, ~~of the project~~ as a function of stimuli received from an environment emulator;

-integrating the software model~~, in a verification mode,~~ into a verification platform, and connecting ~~a~~ the previously validated autonomous circuit emulator in parallel to interfaces of the software model ~~of the circuit under design, and~~ connected to ~~an~~ said environment emulator;

utilizing the verification platform in the verification mode, by comparing response data from interfaces of the software model with the response data structures taken as a reference for the validation of response data transmitted by the software model ~~of the circuit under design~~; and

validating the software model or invalidating the software model by outputting one or more error notifications ~~if an error is detected by said validation~~when the response data of the software model are not identical to the response data structures.

49. (currently amended) A method according to claim 48, ~~further comprising:~~wherein the autonomous circuit emulator generates the response data structures

~~generating~~, ~~using a data processing system and~~ in response to a user input, ~~the autonomous circuit emulator~~ which ~~provides a simulation configuration corresponding~~correspond to the functional specification of the software model of the ASIC ~~using the functional specification~~, the method further comprising;

writing~~, from~~ the functional specification~~,~~ data and storing in ~~a~~ the ~~test~~ verification platform ~~for integrated circuit models~~, a test program for testing the software model of the ASIC, comprising input stimuli sequences to be provided to the software model of the ASIC, related to output stimuli sequences generated by the

autonomous ~~simulation configuration~~circuit emulator, based on the functional specification data;

linking together[[,]] and activating[[,]] the autonomous ~~simulation configuration~~circuit emulator and the ~~test~~ verification platform; and

~~observing the~~comparing output stimuli of both the response data of the software model of the ASIC and the response data structures, wherein the software model of the ASIC is a Hardware Description Language (HDL)-type model, in order to functionally validate ~~the system constituted by~~ the software model of the ASIC ~~and~~ aby the ~~validation~~ test program, and thus validating the software model in ~~comparison~~ relation to the functional specification data.

50. (previously presented) A method according to claim 48, wherein the autonomous circuit emulator communicates with a user to control the activation of previously created and stored models of input stimuli sequences defined in a high-level programming language, and controls the activation of associated programs for the progressive validation of test sequences determined from the models.

51. (previously presented) A method according to claim 48, wherein the functional specification data comprise a sequence of instructions in a low-level programming language, specifying functional models of circuits.

52. (previously presented) A method according to claim 48, wherein the functional specification data are provided in the form of a first specification program in a low level programming language of functional models of circuits, and a second specification program in a high level programming language of functional models of

circuits, and the autonomous circuit emulator performs a co-simulation by synchronizing the execution of the first and second specification programs.

53. (previously presented) A method according to claim 52, wherein the low level language is a Hardware Description Language (HDL)-type and the high level language is C++.

54. (previously presented) A method according to claim 48, wherein the verification platform verifies that the responses of the software model of the ASIC are within response time ranges specified in the functional specification data.

55. (currently amended) A verification platform for on demand verification of a software model of an application specific integrated circuit (ASIC), comprising a data processing ~~means for allowing~~system receiving selection requests from a client to select test models producing input stimuli ~~for~~ applied to the software model of the ASIC, said data processing ~~means being constructed and arranged to read~~system comprising in memory functional specification elements of the ASIC which are read in a high level language and comprising in memory a sequence of programmed instructions ~~that form~~of an emulator program ~~and~~ that generates ~~a functional validation test program constituted by~~ output stimuli of a functional validation test program, ~~from~~ in relation to the input stimuli and the functional specification elements, wherein a comparator compares the output stimuli of the functional validation test program with output stimuli of the software model and ~~that output~~wherein one or more error notifications are output when ~~if an error is detected by said functional validation test program~~compared output stimuli are not identical.

56. (previously presented) A verification platform according to claim 55, further comprising a library of functional models of circuit blocks for a plurality of ASICs and means for selecting models through a definition file of the integrated circuit configuration, for creating a model corresponding to the functional specification of one of said plurality of ASICs that is integrated into the definition of an environment of the ASIC.

57. (previously presented) A verification platform according to claim 55, further including, in a link connecting the platform to a client, first and second serial programming language adaptation circuits, wherein the first serial programming language adaptation circuit transforms commands in a high level programming language used by the client into commands in a low level programming language used by the ASIC model, and wherein the second serial programming language adaptation circuit transforms the commands in the low level programming language back into commands in the high level programming language.

58. (currently amended) A verification platform according to claim 55, further comprising means for executing operations at the same time as the simulation, and, upon detection of an error notification output, ~~means for~~ interrupting operations at the time the error notification appears.

59. (previously presented) A verification platform according to claim 55, wherein the functional specification elements are constituted by a truth table corresponding to the functions of the various functional circuit elements of the ASIC

software model, and further comprising a propagation delay associated with each input and each output pair.

60. (previously presented) A verification platform according to claim 55, wherein the functional specification elements are constituted by a behaviour table corresponding to the functions of the various functional circuit elements of the ASIC software model, and further comprising a propagation delay associated with each input and each output pair.

61. (previously presented) A verification platform according to claim 55, further including a cache memory for storing the blocks used by nodes according to node addresses, and means for managing, for an address used by one or more nodes, a presence vector with one presence indicator per node.

62. (previously presented) A verification platform according to claim 61, wherein the programmed instructions are object-oriented and the emulator is structured as a set of classes for managing a collection of execution hypotheses for a transaction in a memory block of the software model, and for managing transactions that are concurrently colliding using the same memory block.

63. (previously presented) A verification platform according to claim 61, wherein algorithms of the sequence of programmed instructions of the emulator are configured to cause the emulator to perform functions comprising generating predictions, eliminating predictions, readjusting incorrect predictions, reducing the number of valid hypotheses, and terminating collisions.

64. (previously presented) A verification platform according to claim 63, wherein the emulator of the ASIC circuit generates predictions without having to obtain additional information on the internal operation of the ASIC circuit, the ASIC circuit being a circuit under design.

65. (previously presented) A verification platform according to claim 61, wherein the platform is used as an emulator of a router circuit, a circuit with cache or a router circuit with cache.

66. (previously presented) A verification platform according to claim 61, wherein the platform is configured for testing a software model of an integrated circuit (ASIC) on demand and comprises an ASIC emulator for controlling a comparator that receives values generated by a software model of the ASIC circuit tested, upon reception of stimuli sent by at least one stimuli generating circuit storing a test program, an interface for translating the stimuli from an advanced language into a low level language corresponding to that of the software model, and means for validating the verification in case of the detection of a collision by the comparator.

67. (previously presented) A verification platform according to claim 61, further comprising means for selecting the response to stimuli that depend on the composition of the circuits tested, said means for selecting being constituted by a model generated by means for selecting functional models from a library, which associates with each of the models the responses to a given stimulus, the model corresponding to the composition of the circuit to be tested.

68. (previously presented) A verification platform according to claim 67, further including means for storing responses selected so as to create a test model to be applied to the circuit tested during the reception of stimuli.

69. (previously presented) A verification platform according to claim 55, wherein each transaction comprises, at the level of each interface, a request packet and one or more associated response packets, wherein the values of the parameters and/or the transmission time constraints of the request packet and one or more associated response packets can be forced from the functional test program executed by the emulator of the environment, which appropriately translates all of said parameters during the transmission of the request packet and one or more associated response packets to the terminals of the software model of the design.

70. (currently amended) A verification platform according to claim 68, wherein the generation of predictions is performed by the emulator of the circuit ~~without having to obtain additional~~after receiving information on the internal operation of the circuit, the circuit being a circuit under design.

71. (currently amended) A computer-readable medium upon which is embodied a sequence of programmed instructions that, when executed by a processor, cause the processor to perform a method for on demand functional verification of a software model of an application specific integrated circuit (ASIC), wherein said software model is written in a low-level programming language and wherein the

method separately establishes ~~a~~ said software model ~~of the application-specific integrated circuit to be generated~~ and functional verification tests to be applied to the software model ~~of the circuit~~ for constituting a verification platform comprising ~~a transmission~~ an environment emulator debug mode and a verification mode, the method comprising:

creating, in the environment emulator debug~~transmission~~ mode, an autonomous circuit emulator ~~constituted with~~built around a data processing system, receiving in memory functional specification data obtained by replacing, at least by a language adaptation device, the software model ~~which is in a low-level programming language and~~ that physically describes ~~an~~ the ASIC comprising a circuit under design to be validated, with a high level language abstract description, the autonomous circuit emulator generating response data structures in accordance with ~~a~~ said functional specification data, ~~of the project~~ as a function of stimuli received from an environment emulator;

integrating the software model~~, in a verification mode,~~ into ~~a~~ the verification platform, and connecting ~~a~~ the previously validated autonomous circuit emulator in parallel to interfaces of the software model ~~of the circuit under design, and~~connected to ~~an~~ said environment emulator;

utilizing the verification platform in the verification mode, by comparing response data from interfaces of the software model with the response data structures taken as a reference for the validation of response data transmitted by the software model ~~of the circuit under design~~; and

validating the software model or invalidating the software model by outputting one or more error notifications when the response data of the software model are not identical to the response data structures~~if an error is detected by said validation~~.

72. (currently amended) A computer readable medium according to claim 71, further comprising programmed instructions for:

generating, using a data processing system and in response to a user input, the autonomous circuit emulator which provides a simulation configuration corresponding to the software model of the ASIC using the functional specification;

writing~~, from~~ the functional specification data[[,]] and storing in ~~a~~ the ~~test~~ verification platform ~~for integrated circuit models,~~ a test program for testing the software model of the ASIC, comprising input stimuli sequences to be provided to the software model of the ASIC, related to output stimuli sequences generated by the autonomous circuit emulator~~simulation configuration~~, based on the functional specification data;

linking together[[,]] and activating[[,]] the autonomous circuit emulator ~~simulation configuration~~ and the ~~test~~ verification platform; and

~~observing the~~comparing output stimuli of both the response data of the software model of the ASIC and the response data structures, wherein the software model of the ASIC is a Hardware Description Language (HDL)-type model, in order to functionally validate ~~the system constituted by~~ the software model of the ASIC ~~and a~~ by the ~~validation~~ test program, and thus validating the software model in relation~~comparison~~ to the functional specification data.